FIG. 1

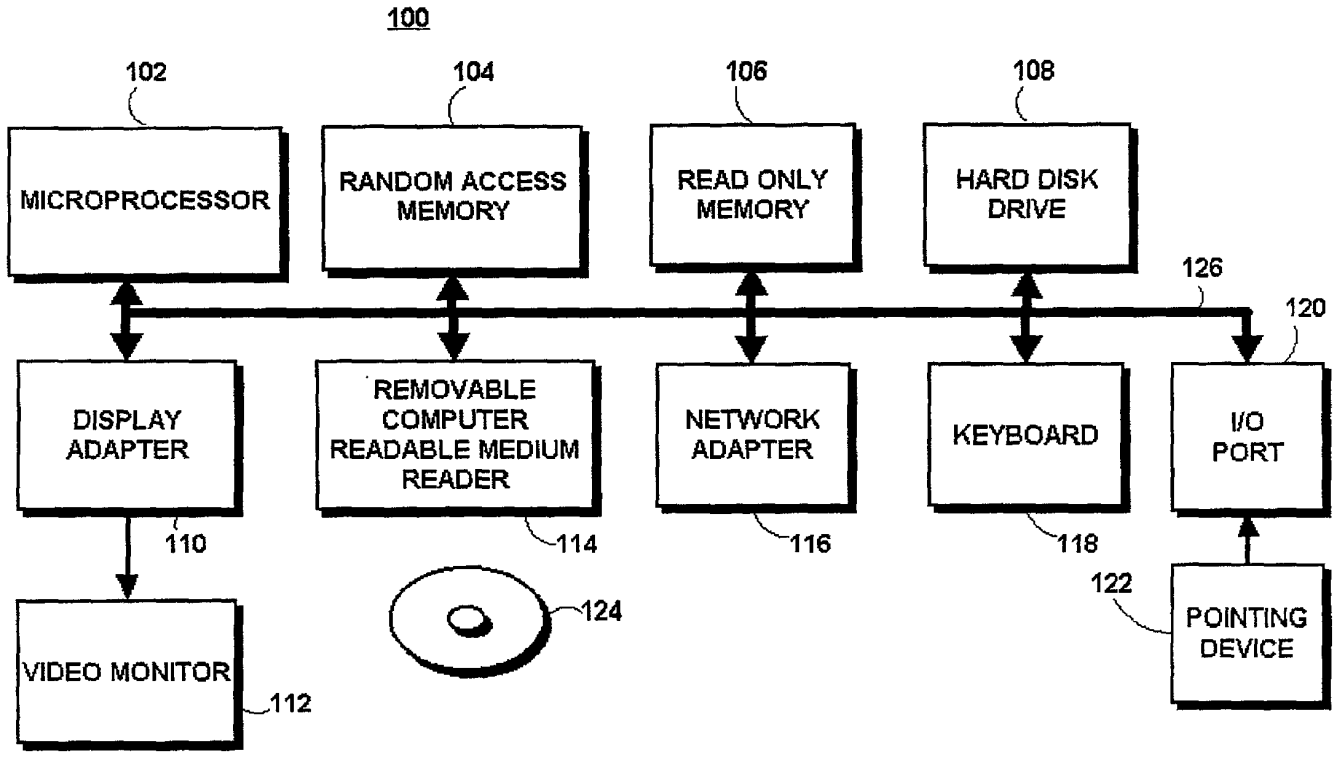

100
102
MICROPROCESSOR
104
RANDOM ACCESS MEMORY
106
READ ONLY MEMORY
108
HARD DISK DRIVE
126
120
DISPLAY ADAPTER
110
VIDEO MONITOR
112
REMOVABLE COMPUTER READABLE MEDIUM READER
114
124
NETWORK ADAPTER
116
KEYBOARD
118
I/O PORT
122
POINTING DEVICE

FIG. 2

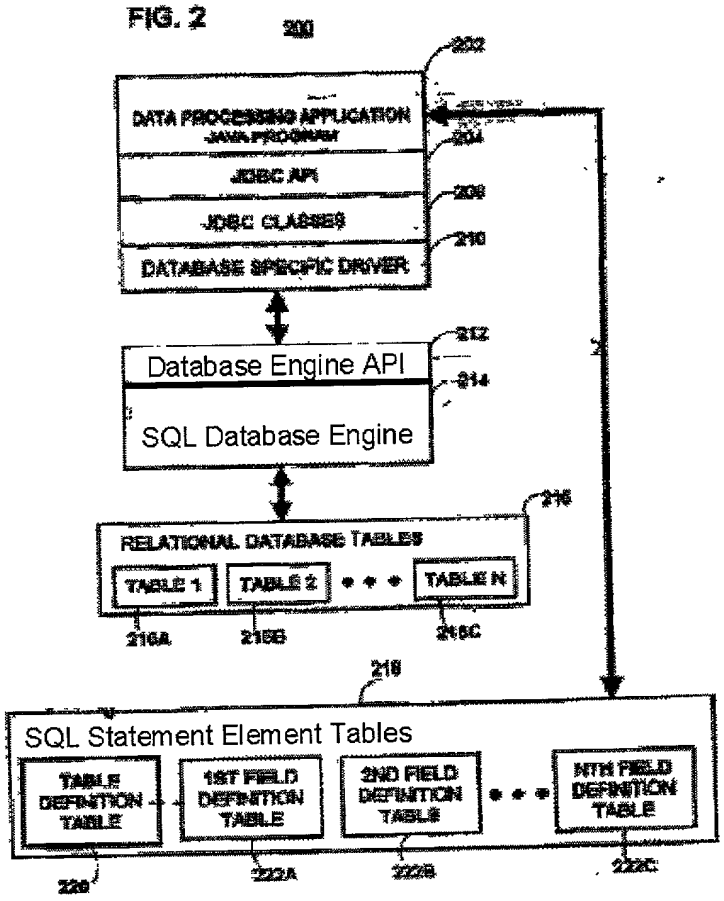
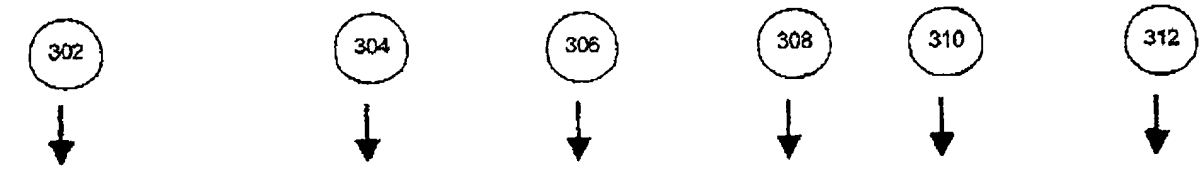

200
202
DATA PROCESSING APPLICATION
JAVA PROGRAM
204
JDBC API
208
JDBC CLASSES
210
DATABASE SPECIFIC DRIVER
212
Database Engine API
214
SQL Database Engine
216
RELATIONAL DATABASE TABLES
TABLE 1
TABLE 2
TABLE N
216A
216B
216C
218
SQL Statement Element Tables
TABLE DEFINITION TABLE
1ST FIELD DEFINITION TABLE
2ND FIELD DEFINITION TABLE
NTH FIELD DEFINITION TABLE
220
222A
222B
222C

FIG. 3

Example Table Definition Table.

300

Expands 220



Target Tablename UTABLENAME ITABLENAME	Query Tablename QTABLENAME	Query Search Clause QSEARCH	Target Action ACTION	Fielddef TableName FTABLENAME	Table Seq # TABSEQ
OperStat	TempStat		Insert	OperStatTab	1
UnitStat	OperStat		Update/Insert	UnitStatTab	2
ComplexStat	UnitStat	ComplexID=1	Update	ComplexTab	3
ComplexStat	UnitStat	ComplexID=2	Update	ComplexTab	3

Docket No. 2000.02.0076

Example Field Definition Tables.

FIG. 4

Field Definition Table: OperStat

400 expands 222A

402 Target Field Name TFIELD	404 Query Select Clause SCLAUSE	406 Query Groupby Clause GCLAUSE	408 Update Set Clause UCLAUSE	410 Update Where Clause WCLAUSE	412 Insert Field Clause ICLAUSE	414 Field Java Type JTYPE	416 Select Field Seq # SFLD SEQ	418 Target Field Seq # TFLDSEQ
Complex	Complex				Complex	String	1	1
Unit	Unit				Unit	String	2	2
Oper	Oper				Oper	String	3	3
Shifts	Shifts				Shift	int	4	4
Calls	Oper				Calls	int	5	5
WrkTime	WrkTime				WrkTime	int	6	6

Field Definition Table. UnitStatTab

420 expands 222B

422 Target Field Name TFIELD	424 Query Select Clause SCLAUSE	426 Query Groupby Clause GCLAUSE	428 Update Set Clause UCLAUSE	430 Update Where Clause WCLAUSE	432 Insert Field Clause ICLAUSE	434 Field Java Type JTYPE	436 Select Field Seq # SFLD SEQ	438 Target Field Seq # TFLDSEQ
Complex	Complex	Complex		Complex=?	Complex	String	1	1
Unit	Unit	Unit		Unit=?	Unit	String	2	2
Calls	SUM(Calls)		Calls=?		Calls	int	3	3
WrkTime	SUM(WrkTime)		WrkTime=?		WrkTime	int	4	4
CumCalls			CumCalls= CumCalls+?		CumCalls	int	3	5
CumWrkTime			CumWrkTime= CumWrkTime+?		CumWrkTime	int	4	6

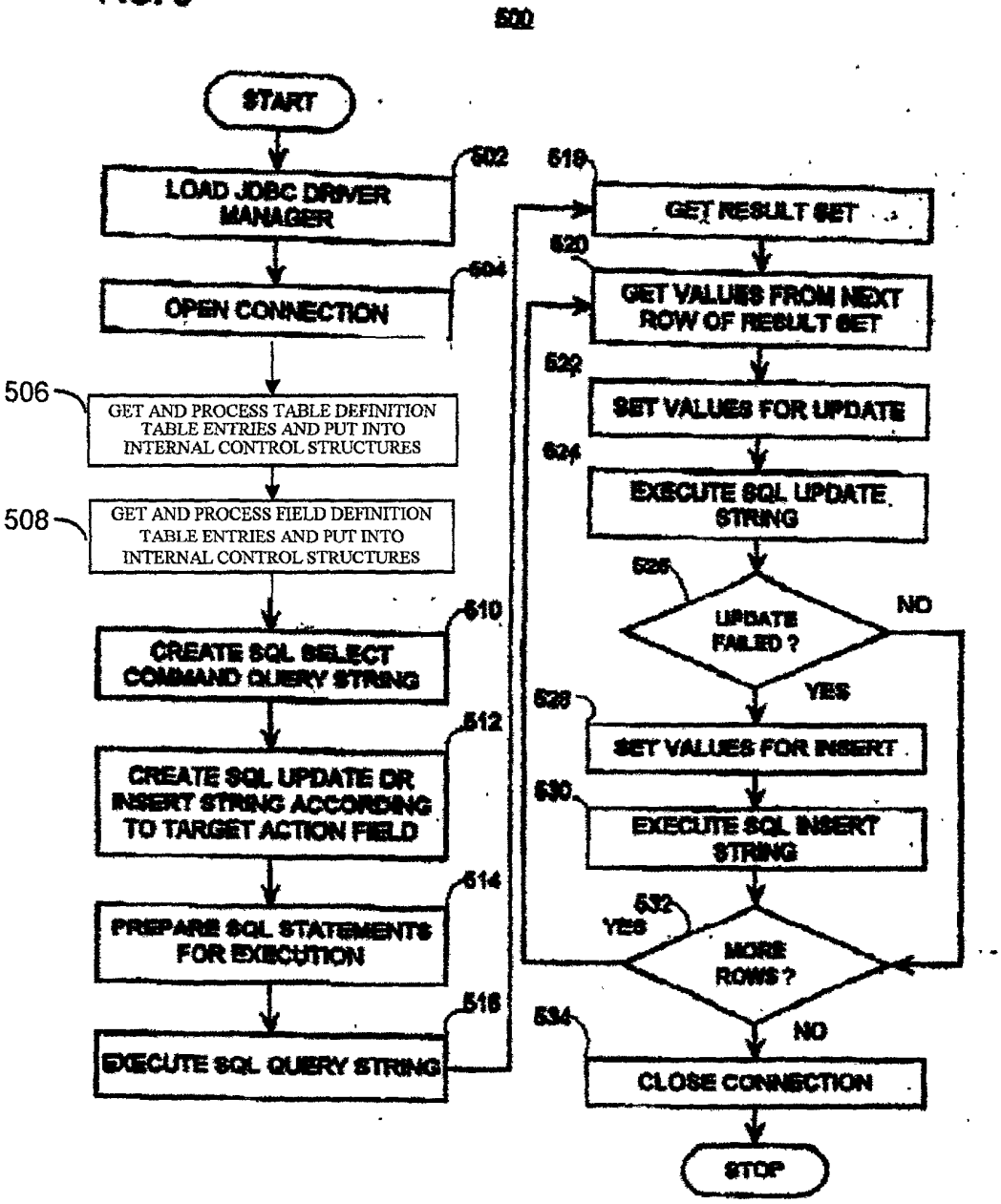
Field Definition Table: ComplexStatTab

440 expands 222C

442 Target Field Name TFIELD	444 Query Select Clause SCLAUSE	446 Query Groupby Clause GCLAUSE	448 Update Set Clause UCLAUSE	450 Update Where Clause WCLAUSE	452 Insert Field Clause ICLAUSE	454 Field Java Type JTYPE	456 Select Field Seq # SFLD SEQ	458 Target Field Seq # TFLDSEQ
Complex	Complex			Complex = ?		String	1	1
Calls	SUM(Calls)		Calls=Calls + ?			int	2	2
HrsWork	Sum(WrkTime)		HrsWork=HrsWork+?			Int	3	3

Docket No. 50277-2000

FIG. 6


500
START
LOAD JDBC DRIVER MANAGER
502
OPEN CONNECTION
504
506
GET AND PROCESS TABLE DEFINITION TABLE ENTRIES AND PUT INTO INTERNAL CONTROL STRUCTURES
508
GET AND PROCESS FIELD DEFINITION TABLE ENTRIES AND PUT INTO INTERNAL CONTROL STRUCTURES
CREATE SQL SELECT COMMAND QUERY STRING
510
CREATE SQL UPDATE OR INSERT STRING ACCORDING TO TARGET ACTION FIELD
512
PREPARE SQL STATEMENTS FOR EXECUTION
514
EXECUTE SQL QUERY STRING
516
GET RESULT SET
518
GET VALUES FROM NEXT ROW OF RESULT SET
520
SET VALUES FOR UPDATE
522
EXECUTE SQL UPDATE STRING
524
UPDATE FAILED ?
526
NO
YES
SET VALUES FOR INSERT
528
EXECUTE SQL INSERT STRING
530
MORE ROWS ?
532
YES
NO
CLOSE CONNECTION
534
STOP

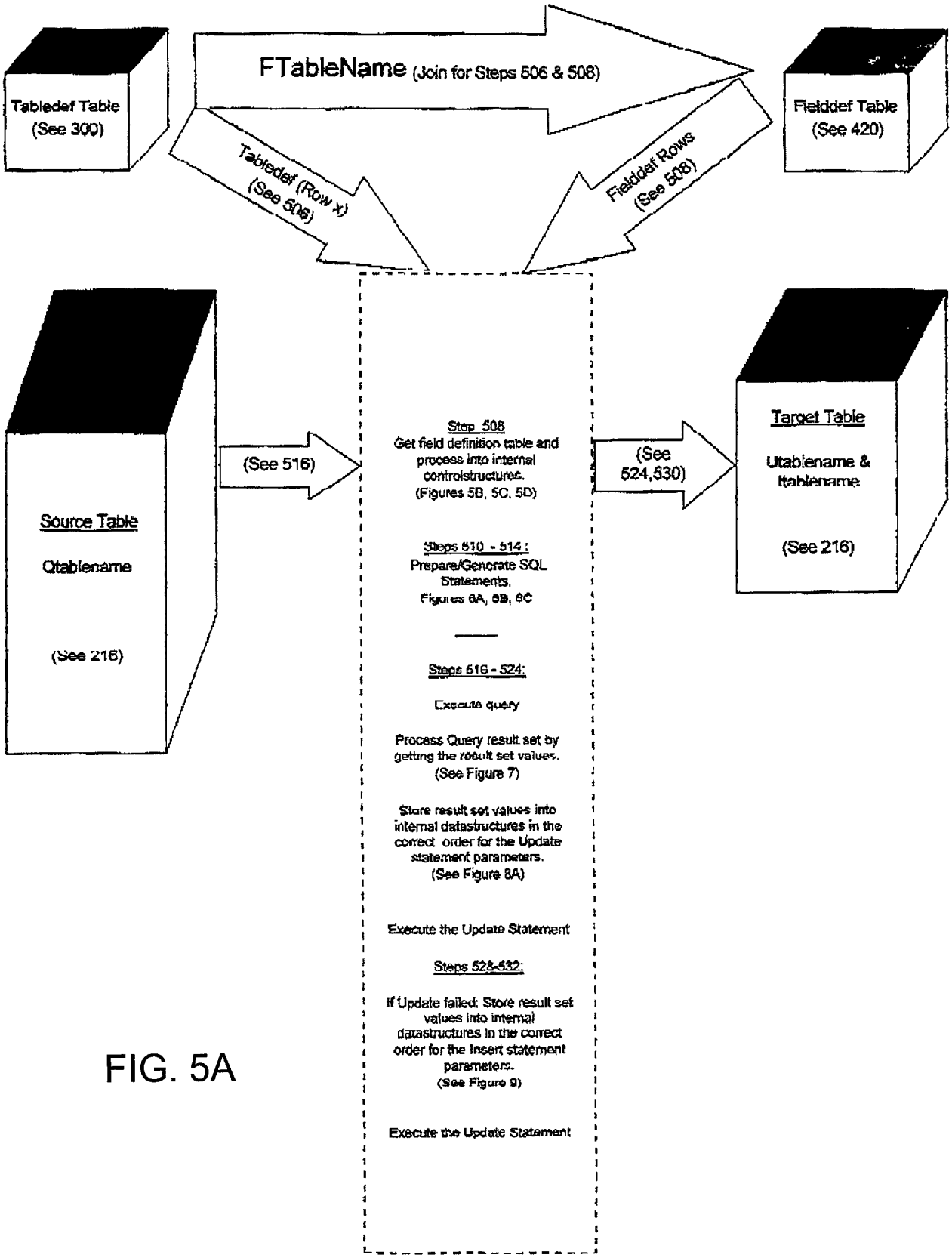


FIG. 5A

Oracle v. Epicrealm - Page 0004

550

```
Select SCLAUSE, JTYPE, SFLDSEQ FROM UnitStatTab WHERE SCLAUSE is NOT NULL ORDER BY SFLDSEQ

i = 1
While not end of resultset
{
// Get values from next row of resultset
SClause[i] = GetString [1]          // get SCLAUSE
SJType[i] = GetString [2]           // get JTYPE
SFldSeq[i] = GetInt [3]             // get SFldSeq
i++
}
Size = i-1;
```

551

FIG. 5B1

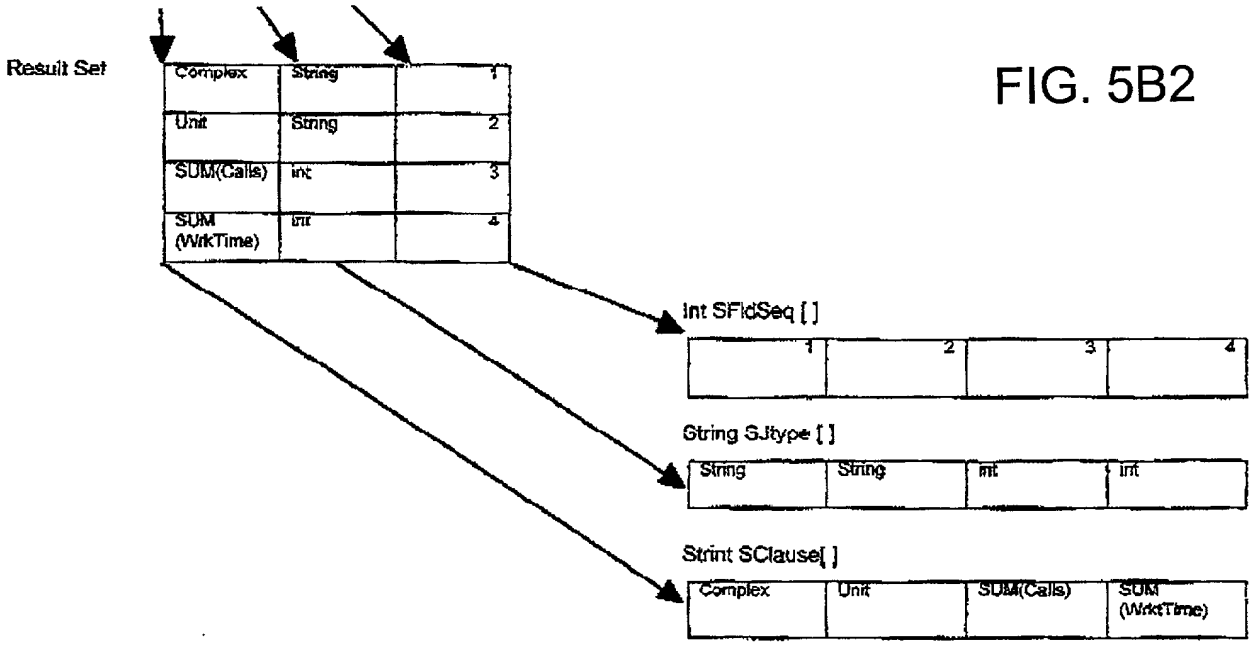


FIG. 5B2

FIG. 5B3

556

```
SELECT  GCLAUSE,SFLDSEQ FROM UnitStatTab WHERE GCLAUSE is NOT NULL ORDER BY SFLDSEQ

i = 1
While not end of resultset
{
// Get values from next row of resultset                    557
GClause[i] = GetString [i]          // get GCLAUSE
i++
}
Gsize=i--
```

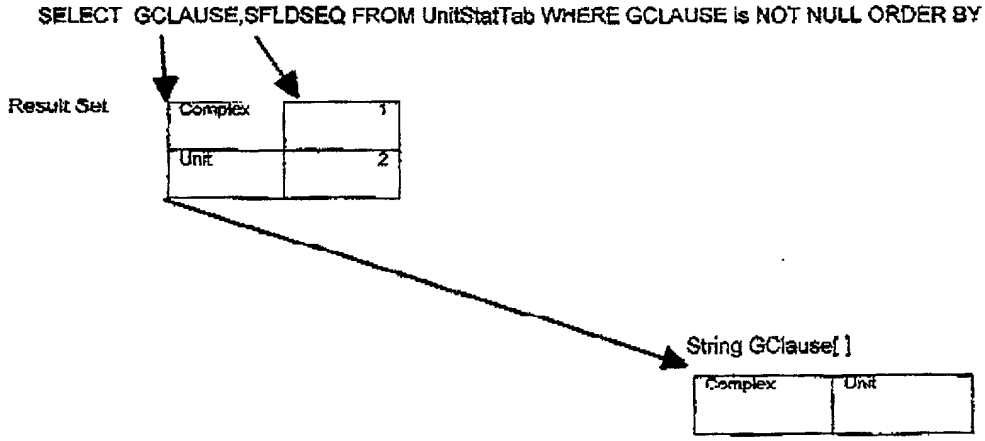


FIG. 5B4

FIG. 5C1

565

Select UCLAUSE, JTYPE, SFLDSEQ,TFLDSEQ FROM UnitStatTab WHERE UCLAUSE is NOT NULL ORDER BY TFLDSEQ

```
i = 1
While not end of resultset
{
// Get values from next row of resultset
UClause[i] = GetString [1]          // get UCLAUSE
UJType[i] = GetString [2]           // get JTYPE
USFldSeq[i] = GetInt [3]            // get SFldSeq
UFldseq[i] = i                      // reset sequence
i++
}
Usize = i--;
```

566

Select UCLAUSE, JTYPE, SFLDSEQ,TFLDSEQ FROM UnitStatTab WHERE UCLAUSE is NOT NULL ORDER BY TFLDSEQ

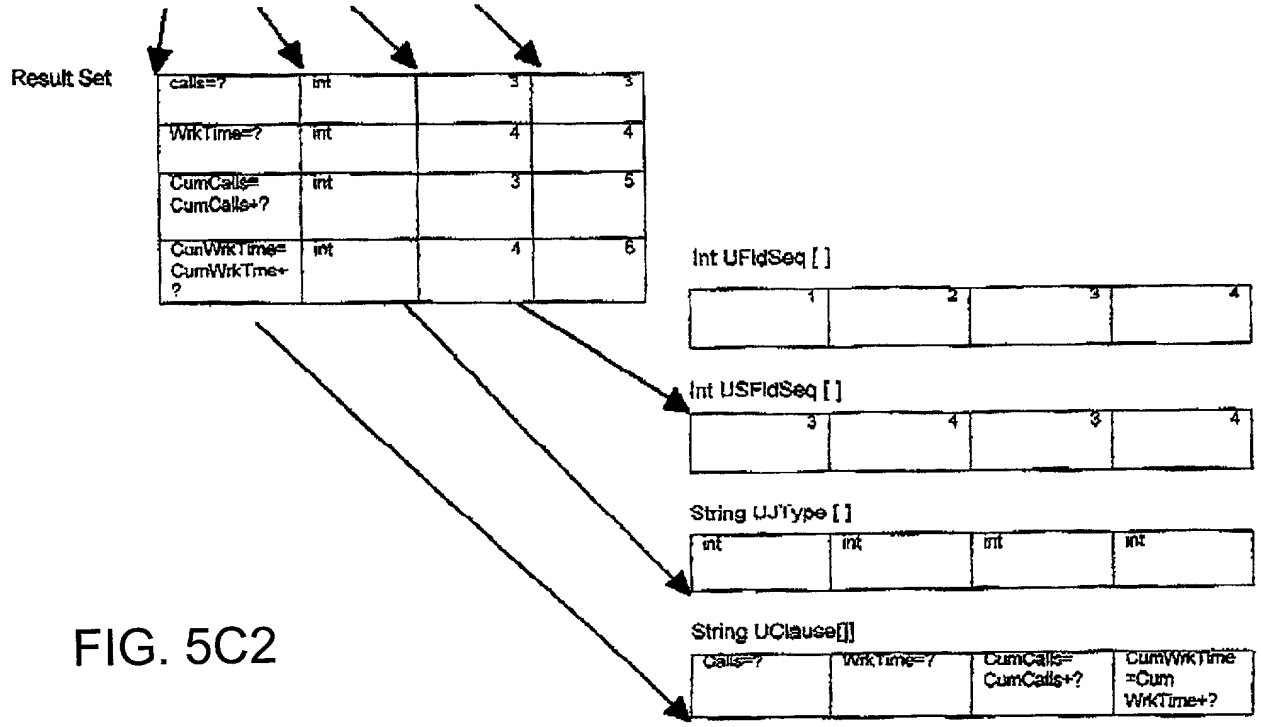


FIG. 5C2

574

```
SELECT WCLAUSE, SFLDSEQ, TFLDSEQ FROM UnitStatTab WHERE WCLAUSE is NOT NULL ORDER BY
TFLDSEQ
```

575

```
i = 1
While not end of resultset
{
// Get values from next row of resultset
WClause[i] = GetString [1]          // get WCLAUSE
WJType[i] = GetString [2]           // get JType
WSFldSeq[i] = GetInt [3]            // get SFldSeq
WFldSeq[i] = Usize + i              // reset sequence
i++
}
Wsize=i--
```

FIG. 5C3

SELECT WCLAUSE, SFLDSEQ, TFLDSEQ FROM UnitStatTab WHERE WCLAUSE is NOT NULL ORDER BY TFLDSEQ

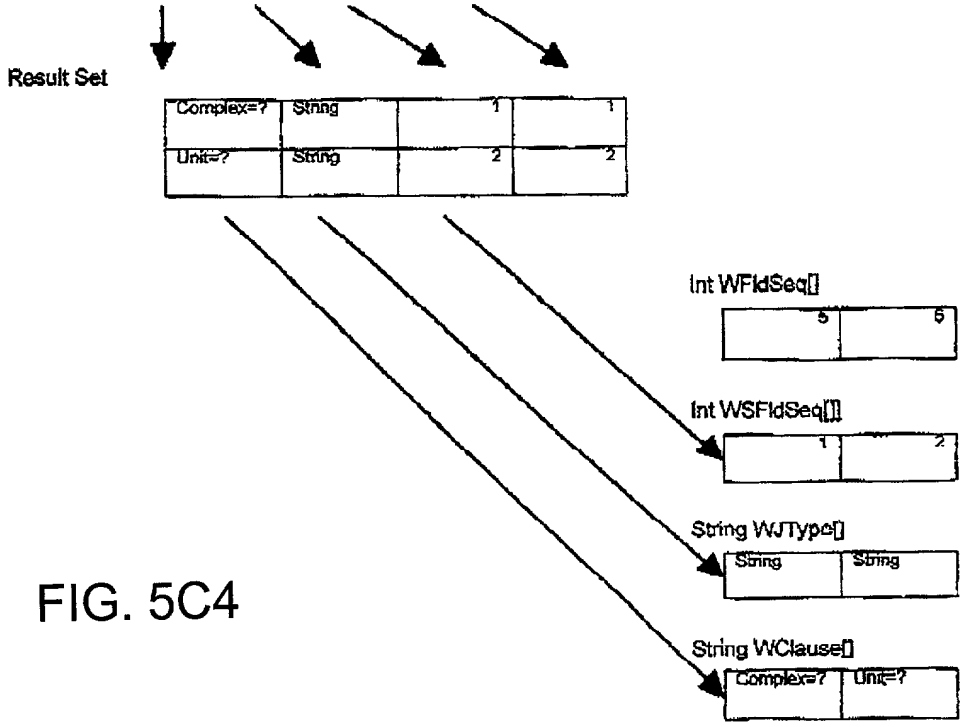


FIG. 5C4

580

Select ICLAUSE, JTYPE, SFLDSEQ, TFLDSEQ FROM UnitStatTab WHERE ICLAUSE is NOT NULL ORDER BY TFLDSEQ

```
i = 1
While not end of resultset
{
// Get values from next row of resultset
IClause[i] = GetString [1]          // get ICLAUSE
IJType[i] = GetString [2]           // get JTYPE
ISFldSeq[i] = GetInt [3]            // get SFldSeq
IFldSeq[i] = i                      // reset sequence
i++
}
isize = i--;
```

581

FIG. 5D1

Select ICLAUSE, JTYPE, SFLDSEQ, TFLDSEQ FROM UnitStatTab WHERE ICLAUSE is NOT NULL ORDER BY TFLDSEQ

Result Set

Complex	String	1	1
Unit	String	2	2
Calls	int	3	3
WrkTime	int	4	4
CumCalls	int	3	5
CumWrkTime	int	4	6

int IFldSeq [ ]

1	2	3	4	5	6

int ISFldSeq [ ]

1	2	3	4	3	4

String IJtype [ ]

String	String	int	int	int	int

String IClause[ ]

Complex	Unit	Calls	WrkTime	CumCalls	CumWrkTime

FIG. 5D2

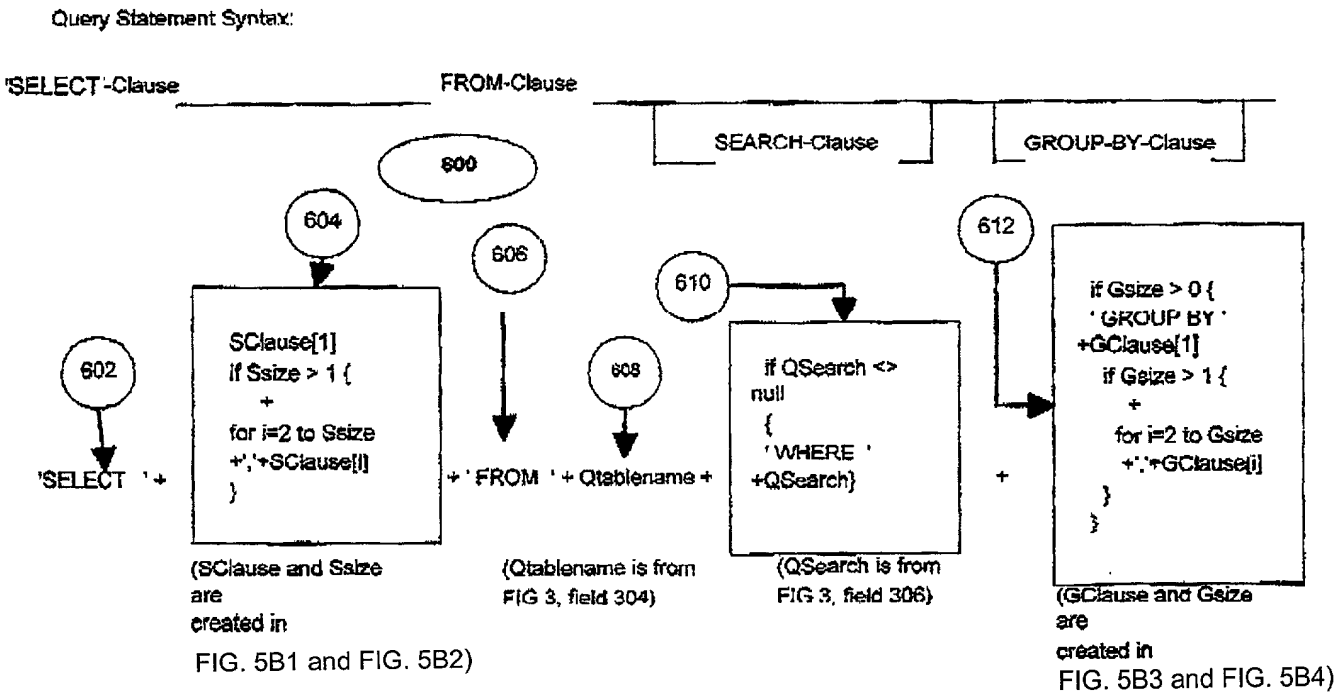


FIG. 6A1

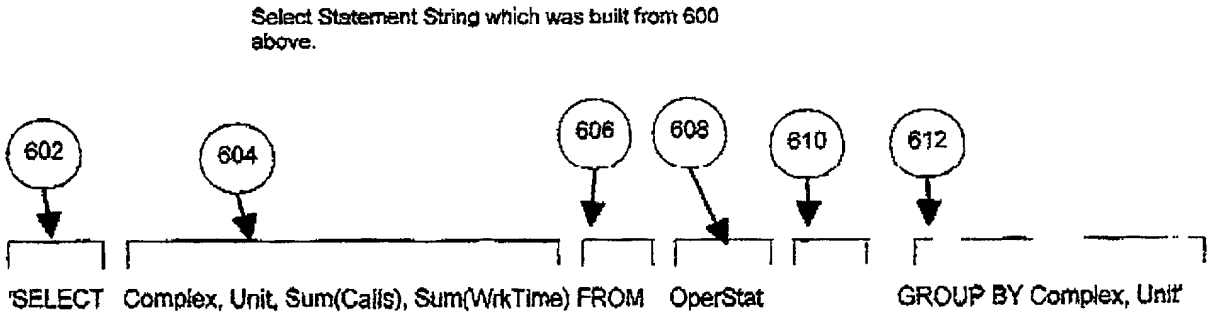


FIG. 6A2

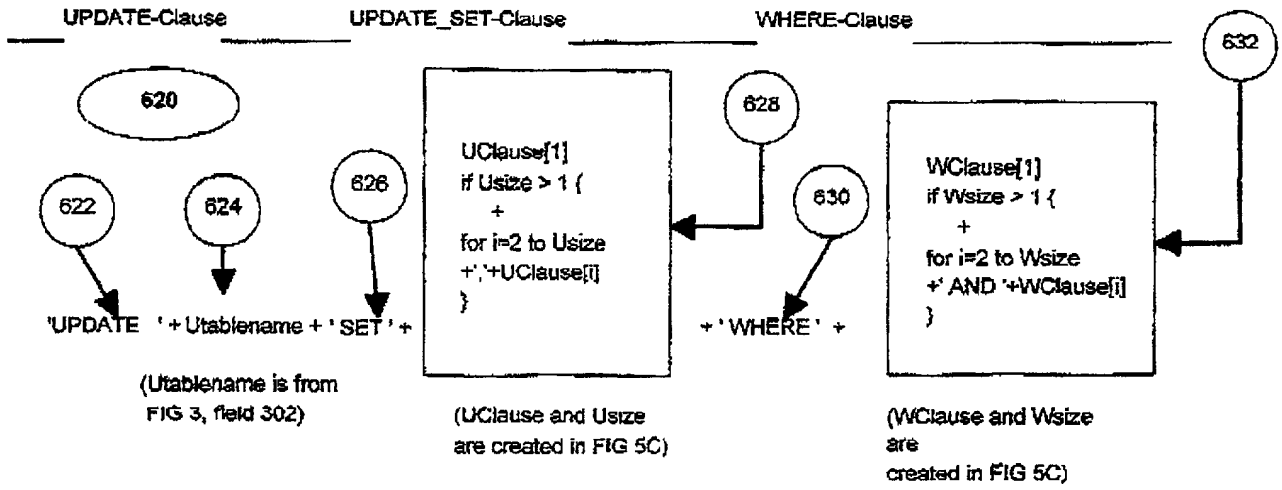


FIG. 6B1

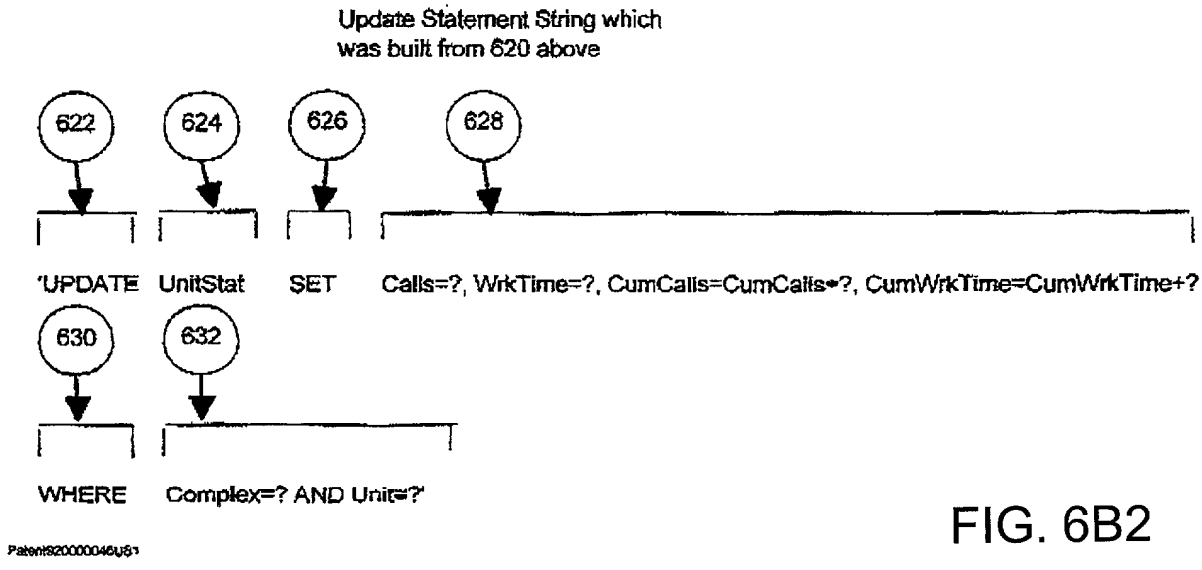


FIG. 6B2

Patent5920000046U51

Insert Statement Syntax:

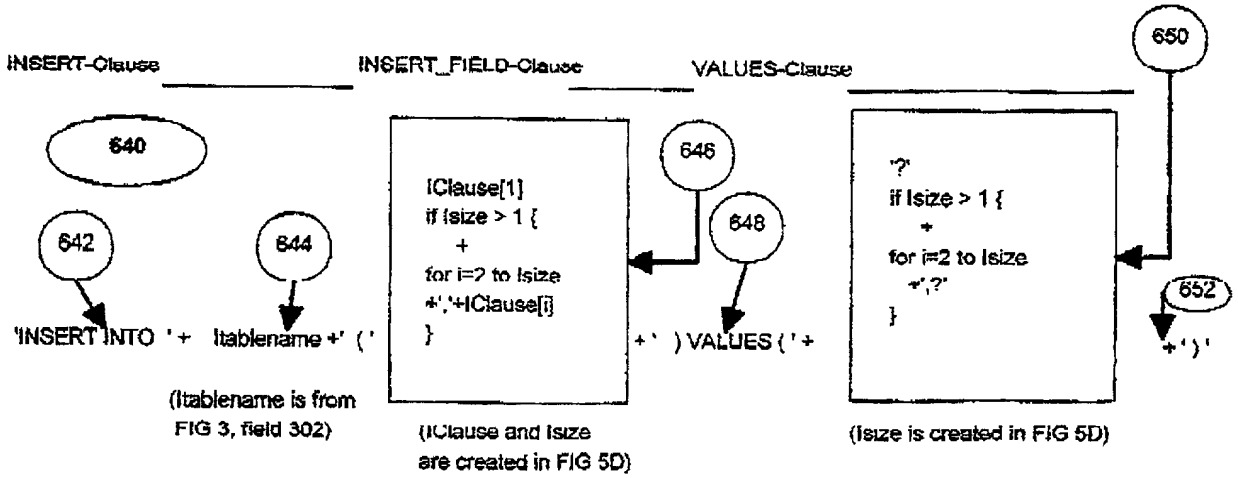


FIG. 6C1

Insert Statement String which was built from 640 above

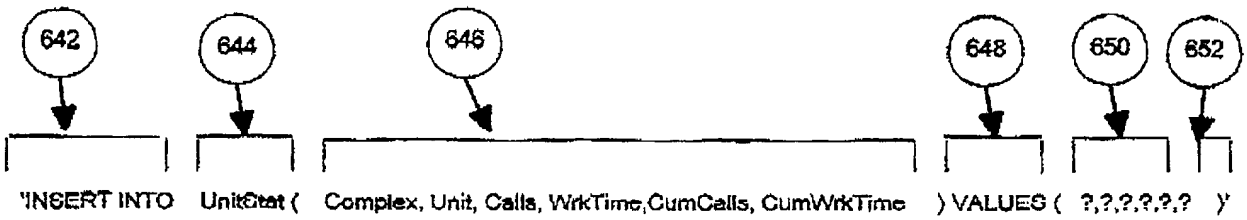


FIG. 6C2

Docket No. 50277-0501

'SELECT Complex, Unit, Sum(Calls), Sum(WrkTime) FROM OperStat GROUP BY Complex, Unit'

```
For i=1 to Ssize
{
  j = SFldSeq[i]
  if SJType[i] = 'String'
     StringRS[j] = GetString[i]
else
     IntRS[j] = GetInt[i]
}
```

Fig. 7A

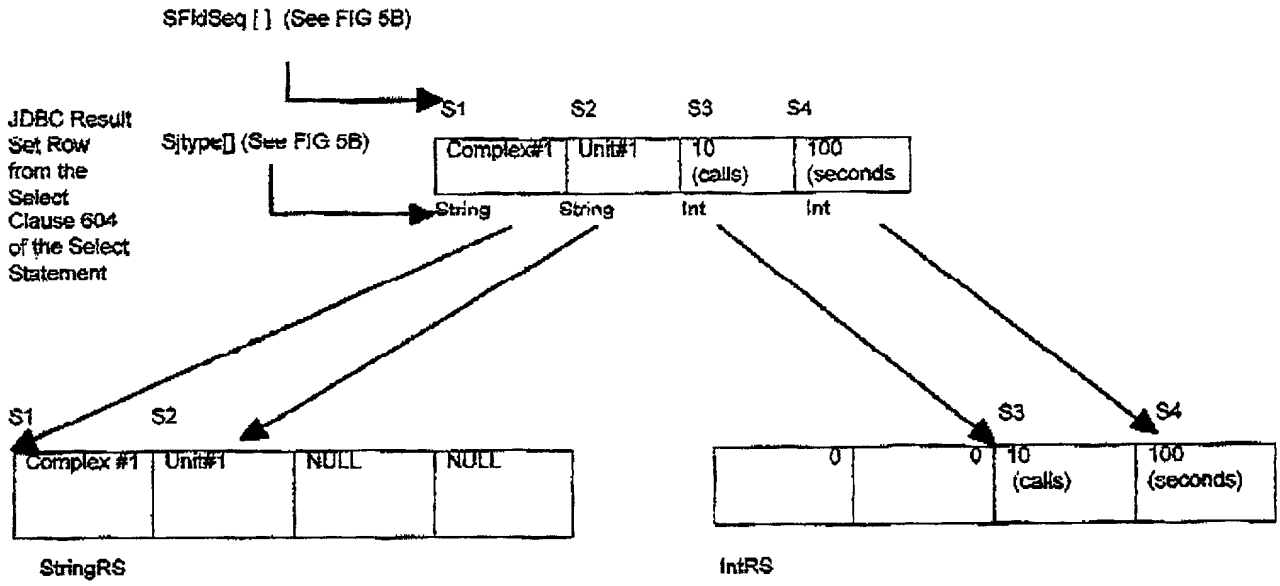


Fig. 7B

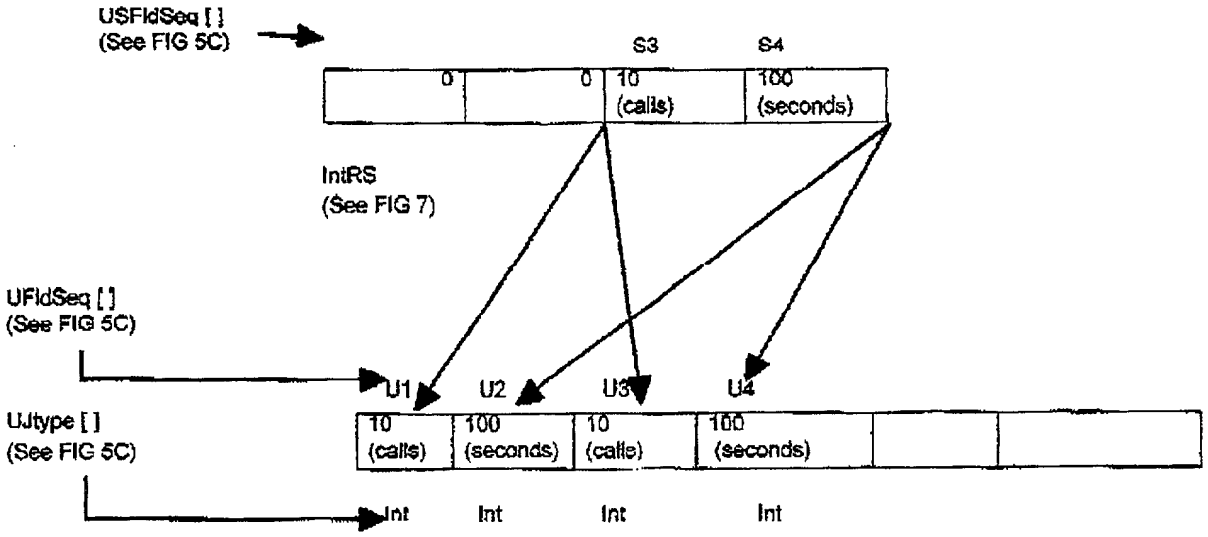
```
'UPDATE  UnitStat    SET    Calls=?, WrkTime=?, CumCalls=CumCalls+?, CumWrkTime=CumWrkTime+?
                              U1          U2                U3                      U4

WHERE    Complex=? AND Unit=?'
                U5          U6
```

```
For i = 1 to Usize
   {
    j= UFldSeq[i]
    k=USFldSeq[i]
    if UJType[i] = 'String'
       setString[ j ] = StringRS[k]
    else
       setInt[j] = IntRS[k]
   }
```

FIG. 8A1

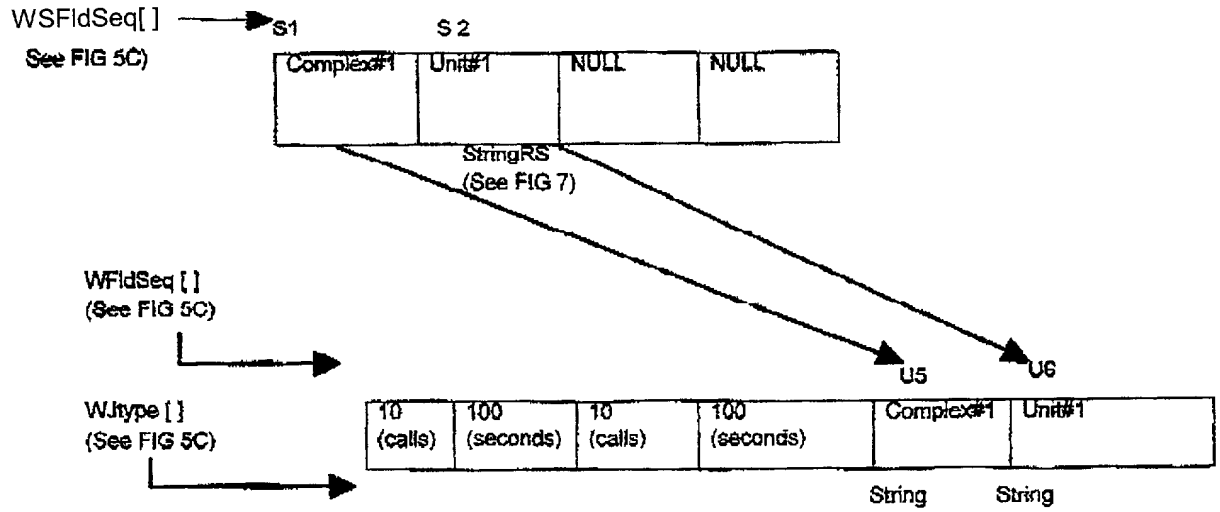
```
For i = 1 to Wsize
   {
    j=WFldSeq[i]
    k=WSFldSeq[i]
    if WJType[i] = 'String'
      setString [j ] = StringRS[k]
    else
      setInt [j] = IntRS [k]
   }
```

FIG. 8A2



JDBC parameters for Update_Set Clause 628 of Update Statement

FIG. 8A3



JDBC parameters for Where Clause 632 of Update Statement

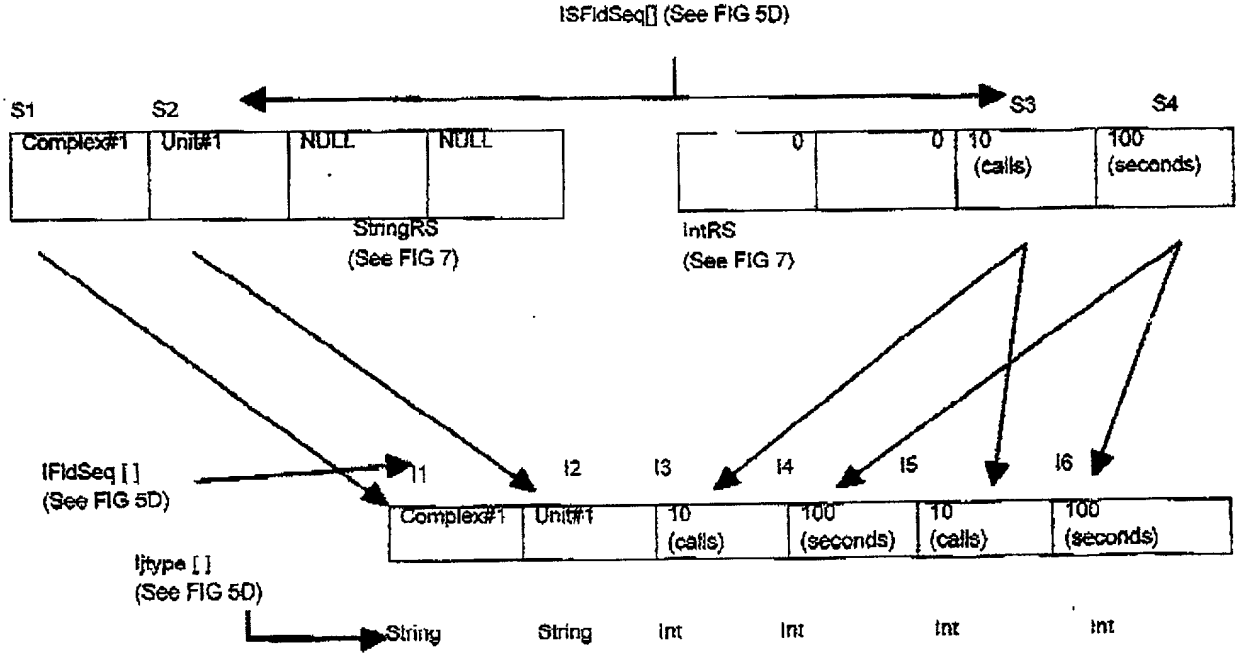
FIG. 8A4

'INSERT INTO UnitStat(Complex, Unit, Calls, WrkTime,CumCalls, CumWrkTime) VALUES (?,?,?,?,?,?)'

I1,I2,I3,I4,I5,I6

```
For i = 1 to lsize
   {
   j=IFldSeq[i]
   k=ISFldSeq[i]
   if IJType[i] = 'String'
      setString[j] = StringRS[k]
   else
      setInt[j]  = IntRS[k]
   }
```

Fig. 9A



JDBC parameters for Values Clause 650 of Insert Statement

Fig. 9B